PC Dot Matrix Display

a minimal user interface

By B. Oehlerking and P. Cohrs

In situations where a PC is used as a dedicated controller the user-interface can often be simplified. Just a few input buttons together with this neat alphanumeric display may be all that is necessary…

The relative low-cost and versatility of today's Personal Computer (PC) and the proliferation of programming tools means that its increasingly more common to find PC's or just PC boards used as the control element in a dedicated application. Whether it is an MP3 player or an industrial controller it is often unnecessary or impractical to use a CRT monitor or LCD panel to display status information. The HIDs (Human Interface Device) for some applications need consist of little more than a few pushbuttons to input and LEDs to output information.

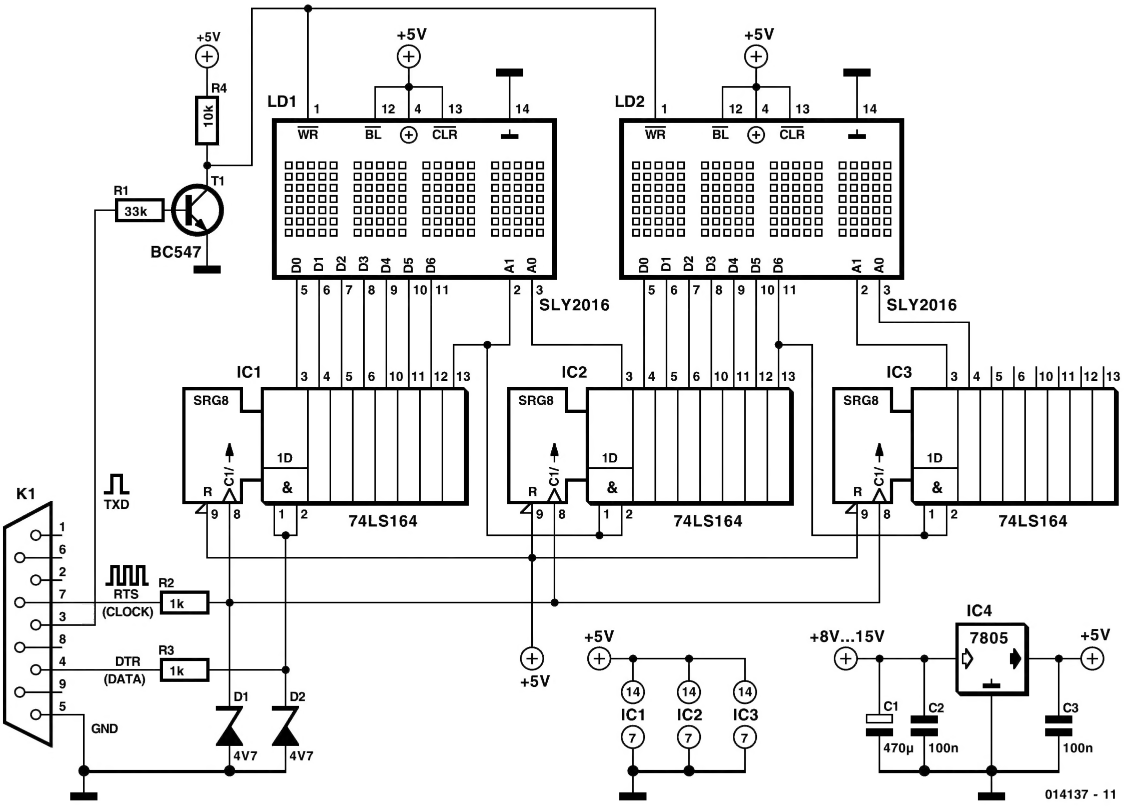


Figure 1. The circuit diagram.

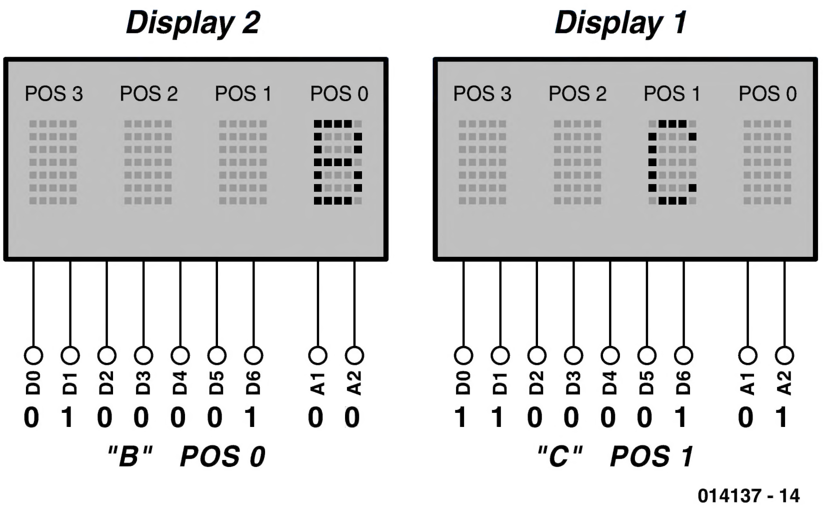
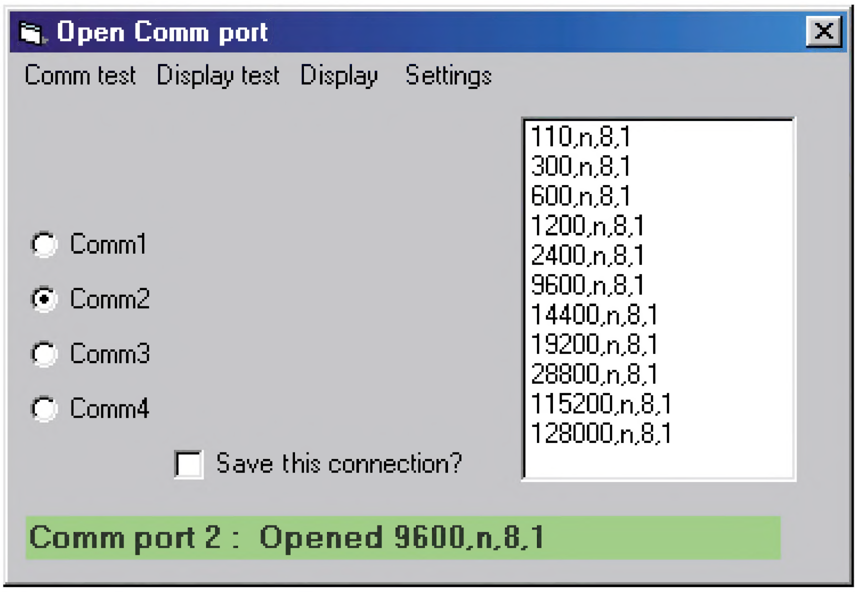


Figure 2. Addressing the displays.



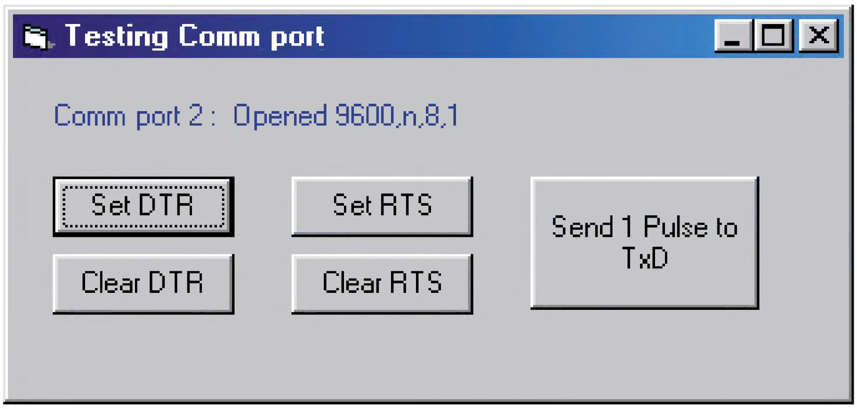
110,n,8,1
300,n,8,1
600,n,8,1
1200,n,8,1
2400,n,8,1
9600,n,8,1
14400,n,8,1
19200,n,8,1
28800,n,8,1
115200,n,8,1
128000,n,8,1

○ Comm1
● Comm2
○ Comm3
○ Comm4

☐ Save this connection?

Comm port 2 : Opened 9600,n,8,1

Figure 3 Port initialisation using the Start module.



Comm port 2 : Opened 9600,n,8,1

Figure 4. Interface test module.

This project describes a simple and economic display solution. A small alphanumeric dot matrix display connecting to the serial COM port of a PC provides a versatile display ideal for many applications. Some typical uses would include:

- Selecting parameters in an embedded application.
- Displaying operating conditions of the PCs processor (CPU clock speed, temperature and Loading etc). The display functions even when the PC monitor is switched off.
- Displaying measurements, counter values date and time or alarm information.

Visual BASIC is used to program the display. The display is connected to a PC using only three signals lines of the serial COM port so that no connection to the PC internals is necessary The display together with the small amount of circuitry can be fitted neatly into a plastic front panel with little difficulty.

The hardware

The display device for this circuit is a four-position LED dot matrix from Siemens (type SLx2016). Each display position consists of a matrix of 7x5 LEDs. The displays are horizontally or vertically stackable. Software enables longer text strings to be displayed in a running text format on the eight-display positions. Software also allows the text speed to be altered from a leisurely jog to a gallop and variably delayed before repeating. The SLx2016 is available in four different display colours. A paragraph giving the most important technical specs on the display accompanies this article.

Figure 1 shows the dot matrix display circuit diagram On-board voltage is regulated to +5 V by IC4. An external mains adapter supplying an output between 8 V to 15 V will be suitable. Each of the digital ICs has a of 0.1 µF capacitor fitted to its supply rails to decouple supply noise. Zener diodes D1 and D2 ensure that the voltage of the serial port signals is kept below Vcc. The displays specified for this circuit are available in red, high efficiency red, green and amber. The displayed characters have good visibility even in conditions of high ambient light and have a much better viewing angle than LCDs.

The SLx2016 has a built-in character generator addressable by seven input signals D0 to D6 allowing up to 2^7 or 128 possible characters to be selected by entering a value in the range from 0 to 127. Two address input signals A0 and A1 select which of the four positions the character will be displayed in. A short pulse on the $\overline{WR}$ input will load the new values of data and address into LD1 and LD2 where they will remain until they are overwritten.

Data to the displays is sent serially over the COM serial port of a PC. Two shift registers type 74LS164 with a serial input are needed to store the control data for one four-position display and three are used for the two displays used here.

An Example

One worked example can often save a half an hour's head scratching. So let's assume that we need to display a capital B in the fifth character position from the right of the display as shown in **Figure 2**. A capital C is already displayed in the second position from the right and should remain after loading the B. The C is in position 1 of display 1 while the B is in position 0 of display 2. The three shift registers IC1, IC2 and IC3 must be loaded with the bit combination shown at the inputs to the displays. The software looks at the bit value furthest to the right and will correspondingly set or reset the DTR port signal (used here as data output). Next it will generate a shift pulse on the RTS signal to latch the data into the first position of the shift register chain. The next data values will now be treated identically until after 18 shift pulses all of the data will be in the correct position. Now a short pulse on the TXD port signal will cause this new data to be loaded and displayed on LD1 and LD2.

The Software

The software for this project is available as a free download from the *Elektor Electronics* website. The number is **014137-11**. Visual BASIC 5.0 was used to write the software and it has been tested under Windows 98 and NT 4.0. The software comprises four modules:

- A start form to open a serial COM port.
- An interface testing form to test the selected port.

Dot matrix display SLx2016

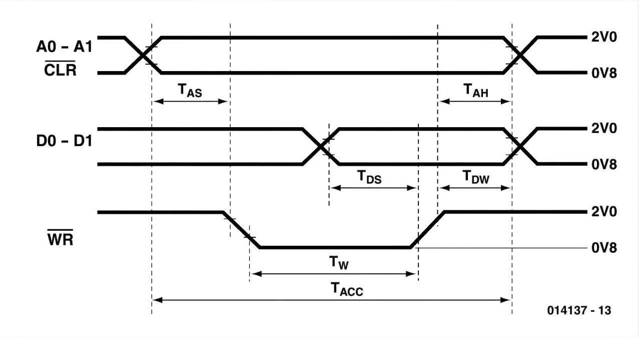
The SLR/SLO/SLG/SLY2016 is a four-digit alphanumeric display with built-in CMOS drivers and decoder circuitry. It is vertically and horizontally stackable (X/Y stackable) and contains an ASCII-Decoder with 128 symbols, a multipexer and LED drivers. The binary code on address pins A1 and A2 selects which of the four display positions the character will be written to. The chip has a total of 14 pins.

Each display position is addressed individually and retains the code that was loaded last until it is overwritten by a new value. The loading process is asynchronous so it can be used by both fast and slow systems. The address and data values must be stable at the input pins before the $\overline{WR}$ signal is pulsed low. The value of the two address bits is latched in on the falling edge of the WR signal while the rising edge latches in the data.

The layout of this display also allows it to be simply mapped to the address space of a processor. Several displays could be stacked together to form a larger display. In this case the data inputs of all the displays are connected in parallel and all the $\overline{WR}$ signals are connected together, this leaves the two address lines to be used for decoding so that the display becomes a write-only area in processor memory.

The $\overline{CLR}$ input can be pulsed low to clear all the stored values and a low on the blanking input $\overline{BL}$ turns the display drivers off without losing the stored data. A pulse-width modulated signal on this pin will control display brightness. Both features are not used in this project so these inputs are tied high.

The built-in character generator uses standard ASCII codes for alphabetic, numeric and punctuation characters but uses all the codes from 00 to 1F (normally non-printing control codes in ASCII) for special characters found in the English, German, Italian, Swedish, Norwegian and Danish languages. The display colour is available in red, high efficiency red, green and amber.

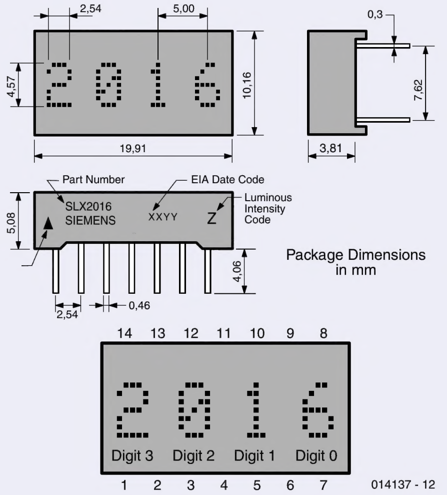


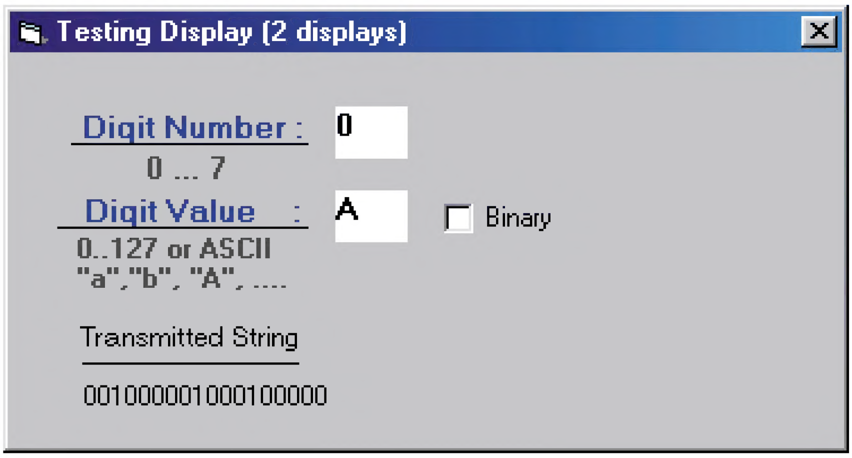
Technical specification:

– Vcc operating voltage:		4.5 V to 5.5 V
– Maximum input voltage with respect to ground (all inputs)		–0.5 to Vcc + 0.5 V
– Storage temperature:		–40 to +85 °C
– Peak Optical wavelength:	Red R	660 nm typ
	Red O (high efficiency)	635 nm typ
	Green G	565 nm typ
	Yellow Y	585 nm typ

Pin outs:

1	$\overline{WR}$ Write
2	A1 Digit Select
3	A0 Digit Select
4	VCC
5	D0 Data
6	D1 Data
7	D2 Data
8	D3 Data
9	D4 Data
10	D5 Data
11	D6 Data
12	$\overline{BL}$ Display Blank
13	$\overline{CLR}$ Clear
14	GND





Digit Number : 0

Digit Value : A

☐ Binary

Transmitted String

0010000001000100000

Figure 5. Display test module.

- A form to input and control the running text.
- A form to send individual characters to the display hardware.

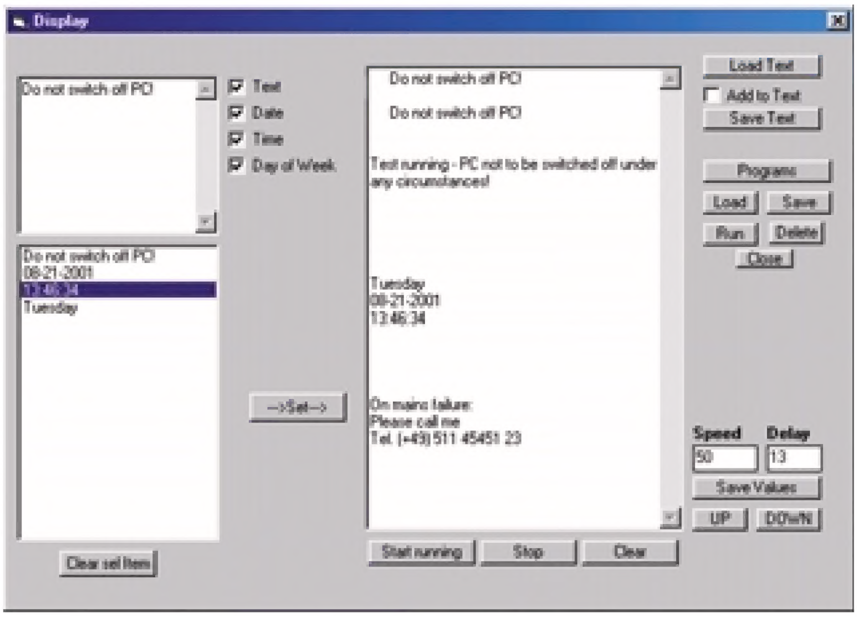
The start module shown in **Figure 3** enables selection of the serial communication port (Com1 to Com4) and its operating parameters or settings. Ticking 'Save this connection?' will save the interface setup to *c:\commport.txt*. Any subsequent re-start of the program will automatically load the last saved configuration. With changes to the menu this module could be adapted for use in other applications that need to configure a serial port.

From the start module a 'Comtest' option can be selected (**Figure 4**). This module displays the actual port configuration set up in the previous module and also allows the signals of the serial port to be tested. The TXD line can be pulsed and the DTR and RTS line can be set high or low. This module needs the 'mod_testcomm.bas' BASIC module in order to function correctly.

Another option from the start module is to check the display with the 'Display test' option (**Figure 5**). This module allows you to position any character or symbol directly on the display. The digit number defines the position while the digit value selects a symbol from the character generator in the display. Entering a value from 0 to 127 will display the corresponding ASCII character. The Basic modules 'mod_testcomm.bas' and 'displ2.bas' are required for this module.

Lastly the start module option 'Display' (**Figure 6**) enables running text messages to be entered. 'Display' comprises three windows. At top left is a small text box, below it is a list box and to the right a large text box showing the message that will be sent to the display. The small text box allows messages to be entered while the list box lists the message together with the standard texts selected by the check boxes. These check boxes have been implemented as an array so it should be relatively easy to modify the source code for your own applications. The Set button transfers the message into the large text box. Along the right hand side of the screen are a series of buttons that perform the following functions:

- Loading, storing and adding text to messages.
- Loading and saving programs in another list box.
- Running the selected program (e.g. to output information for the text box).
- Adjustment of the running text speed and delay before repeating.
- Saving the speed and delay values.
- And finally starting and stopping the running text.



Do not switch off PC!

☑ Text
☑ Date
☑ Time
☑ Day of Week

Do not switch off PC!
08-21-2001
13:46:34
Tuesday

Do not switch off PC!

Do not switch off PC!

Text running - PC not to be switched off under any circumstances!

Tuesday
08-21-2001
13:46:34

On mains failure:
Please call me
Tel. (+49) 511 45451 23

☐ Add to Text

Speed 50

Delay 13

Figure 6. The running text window.

Electromail (*www.rswww.com*) stock the red and green versions of the dot matrix display (SLR2016 and SLG2016 respectively). The quoted price of almost £20 per display seems a tad expensive and might make you think that a cheaper version could be built using simpler single dot matrix displays together with a little additional hardware. Think again! a comparable dot matrix display costs in the region of £2 to £3 per character, multiplying this by four and adding the necessary LED drivers and multiplex circuitry together with the data latches and character generator the final design would not work out any cheaper. The finished product would certainly not be as neat as the design presented here.

(014137-1)